Dynamics Modeling Method Dedicated to System Models with Open- and Closed-loop Structures Subjected to Kinematic and Task Based Constraints

Elżbieta Jarzębowska

Warsaw University of Technology: Politechnika Warszawska

Krzysztof Augustynek

University of Bielsko-Biala: Akademia Techniczno Humanistyczna w Bielsku-Bialej

Andrzej Urbaś (✉ aurbas@ath.bielsko.pl)

University of Bielsko-Biala: Akademia Techniczno Humanistyczna w Bielsku-Bialej

Research Article

Keywords: rigid-flexible systems, constrained dynamics, programmed constraints, automated derivation of dynamical models

Posted Date: January 25th, 2022

DOI: <https://doi.org/10.21203/rs.3.rs-1224916/v1>

License: This work is licensed under a Creative Commons Attribution 4.0 International License.

[Read Full License](#)

Dynamics modeling method dedicated to system models with open- and closed-loop structures subjected to kinematic and task based constraints

Elżbieta Jarzębowska^a, Krzysztof Augustynek^b and Andrzej Urbas^{b*}

^a*Institute of Aeronautics and Applied Mechanics, Warsaw University of Technology, Warsaw, Poland*

^b*Department of Mechanical Engineering Fundamentals, University of Bielsko-Biala, Bielsko-Biala, Poland*

*aurbas@ath.bielsko.pl

Abstract. The paper presents a development of an automated computational procedure for constrained dynamics (CoPCoD) dedicated to derivation of dynamical models of mechanical systems, e.g. manipulators both ground and space or mobile robotic systems, which can be composed of rigid and flexible links. They may be subjected to constraints, which are referred to as programmed, may come from performance requirements, e.g. work or services a system is dedicated to. The CoPCoD structure offers systematic modeling of either open or closed loop constrained structures and results in computationally efficient numerical dynamical equations derivation. The CoPCoD development has its background in the generalized programmed motion equations (GPME) algorithm developed successfully for rigid models of mechanical systems subjected to high order nonholonomic constraints, however, the GPME was not fully automated for computer equation derivations. The two main motivations that underlie the CoPCoD development are to extend the GPME to flexible system models and make it fully automated to computer equation derivations for large classes of systems. The paper presents the general scheme of the CoPCoD architecture and examples which illustrate its application advantages.

Keywords: rigid-flexible systems, constrained dynamics, programmed constraints, automated derivation of dynamical models

1. Introduction

1.1. Background and literature survey

The literature on various aspects of research on rigid and flexible multibody systems is very rich. There is a lot of papers about writing, solving and simulation of dynamic equations of rigid and flexible multibody systems; see e.g. [1,2] and references there. Also, there are multiple papers on designing control techniques for such systems. Depending upon the purpose of modeling, we can distinguish methods aimed to build validation models including nonlinear terms [3,4], methods to derive preliminary design models, which due to their simplicity are attractive to control [5]. The control dedicated models often arise as various versions of linearized equations, small displacement assumptions are adopted for flexible multibody systems and link flexibility is approximated by discretized models using finite element methods or assumed mode methods; see e.g. [6]. The dynamics of the system is usually derived using the Euler-Lagrange formulation. As indicated in [7], there are cases, e.g. in space manipulation during debris capture, when there is a need to have a model for each sub-structure of a system such that it is valid for arbitrary boundary conditions.

Among models that take advantage of the sub-structure approaches, the transfer matrix method gained popularity [8]. In this method the transfer matrix is a relation between the left and right points of state vectors of a flexible body. The state vector at a point is defined as the augmented vector composed of the generalized displacements and the corresponding generalized forces. This method was particularly well suited for modeling structures composed of serially connected bodies or open-chain structures. Later, this approach was merged with the finite-element method and presented as the finite-element transfer matrix method. For example in [9] the Discrete Time Transfer Matrix Method is combined with the Finite Segment Method. It combines the discretization modeling strategy and takes advantages of the computational efficiency of the Discrete Time Transfer Matrix Method.

Some extensions of this method were also developed for control design purposes and for rigid and flexible multibody systems [10]. In [11], the transfer matrix method was applied to closed-loop kinematic chains of flexible bodies. However, this method has some limitations due to the inversion of sub-matrices, which are not always square or invertible, depending on the boundary conditions. It cannot be directly applied to tree-like structures, like a spacecraft composed of a rigid base and flexible appendages on it.

In many works various methods for derivation of linear or linearized models of flexible multibody system dynamics are presented. For example in [7], the approach is based on the two-port model of each body allowing the whole system model to be built by connecting the inputs/outputs of each body model. Boundary conditions of each body are taken into account through inversion of some input–output channels of its two-port model. This one model of the link can be used for a link built into an open or closed-loop mechanism. The method presented in [7] fills the gap between the transfer matrix method and the effective mass-inertia method.

The significant group of dynamic models in engineering applications originates from the multibody dynamics and control of rigid and flexible systems for which Newton-Euler and Lagrange based dynamics derivation methods are used. Among them, there are special methods dedicated to constrained systems in both analytical and computational approaches. As reported in [12], constrained system dynamical models constitute a significant class of models essential to engineering applications. They are used in so many applications, e.g. in motion analysis in robotics for multi-link industrial manipulators and robots, for ground, underwater, aerospace and space vehicle system dynamics, control, and performance analysis for variety of their missions, that computational methods for derivation of their dynamical models, and their solution methods, became separate research areas, see e.g. works [13, 14] and references there. A lot of application examples for multibody system dynamics models can be found in [15, 16]. Reviewing derivation methods of dynamic models for mechanical systems, the first observation is that they are for systems subjected to position and first order constraints, which are material ones; see e.g. [17 - 19] and references there. The task based constraints, control and performance constraints and other requirements available in equation forms are not merged into these models. Consequently, these models are derived using classical analytical dynamics methods, i.e. based upon the Lagrange approach and its modifications. Another approach is to dynamics modeling and control approach is based upon the framework developed by Udwadia and Kalaba [20 - 23]. This is an elegant apparatus for deriving constrained dynamics based upon the classical mechanics approach and it serves systems subjected to position and kinematic constraints. Also, there are many specialized computational packages for generation of constrained system dynamical models. Generally, they are either Newton-Euler equations or Lagrange equations based and dedicated to rigid multibody systems; see e.g. [24, 25]. The consequence of the derivation method applied to obtain constrained system models is that the resulting equations of motion may not be convenient for their final applications, e.g. they contain Lagrange multipliers which need to be eliminated for most model based controller designs. The second observation is that some of the proposed methods of motion equations

derivation are specific and may be challenging for applications for systems with flexible components; see e.g. [26]. The good illustrations of the two observations mentioned above are works [27] – [29]. These works have had significant impact upon the insight into the constrained system structures and for the subsequent control design, like [29] where geometric properties of dynamic models were shown to be the key to model properties from stability and control points of view. However, constraint kinds considered there and results presented were for material constraints, position and kinematic ones. Also, new challenges in dynamic modeling and control designs emerged from interdisciplinary or strongly dedicated models as well as requirements for system performance or their service mission demands. The new challenges were brought, for example, by underwater vehicles and spacecraft, see e.g. [30]. For the latter ones, long and lightweight structures require link flexibility incorporation into dynamic models as well as mission service and control requires formulation of task based constraints and some of them can be easily formulated by constraint equations. Usually such constraints, if formulated, are position levels; see e.g. [31, 32]. Also, one more challenge arises for spacecraft modeling and simulation; this is to keep conserved quantities of motion like the linear and angular momentum. The linear and angular momenta can be considered the kinematic constraints and incorporated into constrained dynamics [33].

1.2. Formulation of the problem of interest for this investigation

In most dynamics derivation methods, of either approaches, systems subjected to position constraints are considered. In some cases first order constraints are incorporated in the Lagrange setting. Systems with other constraints are not handled. There is a group of constraints, which are generated by designers, control engineers or system operators that have to be executed during a system operation. Examples can be found in, e.g. [34 - 38]. For systems subjected to such constraints the preliminary results for obtaining the computationally efficient dynamical models were presented in [12, 39, 40]. There, constraints of this kind, referred to as programmed, at the position level, were incorporated into a rigid-flexible system dynamics for open loop structures.

The analytical dynamics background for the dynamic models are works [41, 42], where the generalized programmed motion equations (GPME) are derived. The GPME proved to be effective to derive constrained dynamics, referred to as reference dynamics, for rigid system models in various parameter settings like generalized or quasi-coordinates. The GPME were employed to design a control strategy architecture for tracking predefined motions. Also, preliminary research results in application of the GPME approach to variable mass and configuration systems are promising [43, 44]. However, The GPME depend heavily upon analytical dynamics calculations what is not convenient to multi-degree of freedom systems whose dynamics require computation.

1.3. Motivation and contribution of this study

This paper is motivated by two factors, being quite critical in designing the constrained dynamics and controllers for programmed constraints execution. The first one, as mentioned above, is the specification of programmed constraints as outcomes of some work regimes, e.g. velocity of a manipulator end effector has to be modulated during the manipulator work. From the manipulator work planning and control points of view, ability of this constraint incorporation into the system dynamics can be advantageous. The second motivation comes from variety of mechanical system structures, e.g. composed of rigid and flexible parts with open- and closed-loop structures, and one modeling method that encompasses these systems would be welcome. Finally, it is the authors premise that dynamics modeling should be

proactive not reactive what means that its dedication should be clear before but not after modeling. It is well described in [45], where based upon engine dynamics examples advantages and disadvantages of the two approaches are presented.

The main contribution of the paper is development of the automated computational procedure for constrained dynamics (CoPCoD) of the flexibly supported systems. In contrast to the authors' earlier work, programmed constraints are defined as reonomic. The presented approach is based on the GPME and preserves its properties, however is fully computationally automated. The basic one is that the constrained dynamics it produces is in the reduced state form, i.e. constraint reaction force are eliminated at the derivation level but not afterwards as in the Lagrange approach, which is widely used to derive constrained dynamics. This is the essential advantage of the CoPCoD which may serve directly for constrained motion analysis as well as for control design. The CoPCoD offers automation of the constrained dynamics generation for rigid and flexible system models with open- and closed-loop structures, and possibility of incorporation into these dynamics any position or first order material or non-material constraints, including these related to system operation regimes.

The paper presents the general structure of the CoPCoD and demonstrates the ease of its application in cases of complex kinematic structures, e.g. a tree structure with open and closed loop chains. Examples illustrate how the CoPCoD structure supports modeling and computation of complex structures.

1.4. Organization of the paper

The paper is organized as follows. After presenting the background and motivation of the paper, the general algorithm of CoPCoD for open- and closed-loop serial kinematic chains subjected to programmed constraints is presented in Section 2. In this section, models of a flexible base support, a flexible link and friction in joints are described. Further, in Section 3, the results of numerical simulations for two model examples representing open loop kinematic chain (OLKC) and closed loop kinematic chain (CLKC) structures are discussed. In simulations, the influence of joint and link imperfections, base support flexibility and material and programmed constraint forms on the reference dynamics are analyzed. The paper closes with conclusions and the list of references.

2. Architecture of the automated Computational Procedure for Constrained Dynamics (CoPCoD)

To develop a general structure of the automated Computational Procedure for Constrained Dynamics, referred to as the CoPCoD, let us consider systems of general structures that can be composed of open-loop kinematic chains (OLKC) (Fig. 1a) and closed-loop kinematic chains (CLKC) (Fig. 1b). Both kinds of systems are attached to the ground in $n_e \times n_{sde_s}^{(i)}$ points by means of spring-damping elements (sde_s), where n_e is the number of layers and $n_{sde_s}^{(i)}$ is the number of sde_s in each layer.

Fig. 1. Multibody systems with the flexibly supported base: a) OLKC system b) CLKC system.

The cut-joint technique is used when an analyzed structure consists of CLKC.

The kinematic chain α is built of $n_l^{(\alpha)}|_{\alpha \in \{c, c_1, c_2\}}$ rigid and $n_{fl}^{(\alpha)}|_{\alpha \in \{c, c_1, c_2\}}$ flexible links. Flexible links are discretized by means of the Rigid Finite Element Method in the modified formulation [46, 47].

2.2.1. Generalized coordinates and homogeneous transformation matrices

Motion of a system can be defined by a vector of generalized coordinates as follows:

$$\mathbf{q} = (q_v)_{v=1, \dots, n_{dof}} = \begin{cases} \begin{bmatrix} \mathbf{q}^{(b)^T} & \mathbf{q}^{(c)^T} \end{bmatrix}^T & \text{for OLKC,} \\ \begin{bmatrix} \mathbf{q}^{(b)^T} & \mathbf{q}^{(c_1)^T} & \mathbf{q}^{(c_2)^T} \end{bmatrix}^T & \text{for CLKC,} \end{cases} \quad (1)$$

where: n_{dof} is number of generalized coordinates which describe a system motion,

$$n_{dof} = \begin{cases} n_{dof}^{(b)} + n_{dof}^{(c)} & \text{for OLKC,} \\ n_{dof}^{(b)} + n_{dof}^{(c_1)} + n_{dof}^{(c_2)} & \text{for CLKC,} \end{cases}$$

$n_{dof}^{(b)}, n_{dof}^{(\alpha)}|_{\alpha \in \{c, c_1, c_2\}}$ is number of generalized coordinates of the base and chain α .

$\mathbf{q}^{(b)}$ is a vector defining motion of flexibly supported base b ,

$\mathbf{q}^{(\alpha)}|_{\alpha \in \{c, c_1, c_2\}}$ is a vector describing motion of chain α with respect to base b .

In the case, when the system reference motion is analyzed, i.e. according to constraints imposed upon it, the vector of generalized coordinates is divided into independent and dependent ones, i.e.

$$\mathbf{q} = (q_v)_{v \in i_c \cup i_{d_c}} = \mathbf{q}_{i_c} \cup \mathbf{q}_{d_c}, \quad (2)$$

where i_c, i_{d_c} are indices of the independent and dependent coordinates, respectively.

Motion of a link $(\alpha, i)|_{\alpha \in \{c, c_1, c_2\}}$ with respect to the global reference frame $\mathcal{F}^{(0)}$ is described by the following vector:

$$\mathbf{q}^{(\alpha, i)}|_{\substack{\alpha \in \{c, c_1, c_2\} \\ i=1, \dots, n_l^{(\alpha)}}} = \begin{bmatrix} \mathbf{q}^{(b)^T} & \mathbf{q}^{(\alpha, i-1)^T} & \tilde{\mathbf{q}}^{(\alpha, i)^T} \end{bmatrix}^T, \quad (3)$$

where: $n_l^{(\alpha)}$ is the number of links (rigid and flexible) in a chain α , $n_l^{(\alpha)} = n_{rl}^{(\alpha)} + n_{fl}^{(\alpha)}$,

$\tilde{\mathbf{q}}^{(\alpha, i)}$ is a vector defining motion of a link (α, i) with respect to a preceding link $(\alpha, i-1)$ and $\mathbf{q}^{(\alpha, 0)} \in \emptyset$.

When the link (α, i) is treated as flexible, its motion with respect to the preceding link is described by a vector:

$$\tilde{\mathbf{q}}^{(\alpha, i)}|_{\substack{\alpha \in \{c, c_1, c_2\} \\ i=1, \dots, n_{fl}^{(\alpha)}}} = \begin{bmatrix} \tilde{\mathbf{q}}^{(\alpha, i, 0)^T} & \dots & \tilde{\mathbf{q}}^{(\alpha, i, r)^T} & \dots & \tilde{\mathbf{q}}^{(\alpha, i, n_{rfe}^{(\alpha)} - 1)^T} \end{bmatrix}^T, \quad (4)$$

where $\tilde{\mathbf{q}}^{(\alpha, i, r)}$ is a vector defining motion of rfe (α, i, r) with respect to the preceding rfe $(\alpha, i, r-1)$.

Homogeneous transformation matrices from a local frame of a link to the global reference frame $\mathcal{F}^{(0)}$ can be obtained in the following way:

$$\mathbf{T}^{(\alpha, i)}|_{\substack{\alpha \in \{c, c_1, c_2\} \\ i=1, \dots, n_l^{(\alpha)}}} = \mathbf{T}^{(b)} \prod_{j=1}^i \tilde{\mathbf{T}}^{(\alpha, j)}, \quad (5)$$

where: $\mathbf{T}^{(b)}$ is the transformation matrix from a frame $\mathcal{F}^{(b)}$ fixed at the flexibly supported base to the global reference frame $\mathcal{F}^{(0)}$ and $\tilde{\mathbf{T}}^{(\alpha,j)}$ is the transformation matrix from a local frame of the link (α,i) to a local frame of a preceding link in a kinematic chain.

The homogeneous transformation matrices of rfses from their local frames $\mathcal{F}^{(\alpha,i,r)}$ to the global reference frame $\mathcal{F}^{(0)}$ are determined by:

$$\mathbf{T}^{(\alpha,i,r)} \Big|_{\substack{\alpha \in \{c, c_1, c_2\} \\ i=1, \dots, n_i^{(\alpha)} \\ r=0, \dots, n_{rfe}^{(\alpha,i)}-1}} = \begin{cases} \mathbf{T}^{(b)} \prod_{j=0}^r \tilde{\mathbf{T}}^{(\alpha,i,j)} & \text{if } i=1, \\ \mathbf{T}^{(\alpha,i-1)} \prod_{j=0}^r \tilde{\mathbf{T}}^{(\alpha,i,j)} & \text{otherwise.} \end{cases} \quad (6)$$

2.2. Generalized Programmed Motion Equations (GPME) for systems subjected to position and velocity constraints

The constrained dynamics, referred to as the reference dynamics, is obtained based upon a dedicated method to constrained dynamics generation, i.e. upon the Generalized Programmed Motion Equations (GPME). The method developed for constrained rigid system structures subjected to arbitrary order nonholonomic constraints, which can specify motion demands and are called programmed, is detailed in [41, 42]. It enables verification of the system behavior when it is subjected to programmed constraints, e.g. vibrations that may accompany the desired motions specified by these constraints. It enables then defining and analyzing desired motions for performing servicing tasks. As detailed in [41, 42], the constraint equations imposed as control goals on system performance or as service tasks and can be presented in a general form as:

$$\mathbf{B}(t, \mathbf{q}, \dot{\mathbf{q}}, \dots, \mathbf{q}^{(p-1)}) \mathbf{q}^{(p)} + \mathbf{s}(t, \mathbf{q}, \dot{\mathbf{q}}, \dots, \mathbf{q}^{(p-1)}) = \mathbf{0}, \quad (7)$$

The constraints can be material for $p=0,1$, or programmed for $p \geq 1$. The programmed constraints are imposed by a designer or a control engineer to obtain a system desired performance, e.g. they can be imposed upon acceleration $p=2$ or jerk $p=3$, as well as for desired trajectories with $p=1$. The constraints form (7) is the generalized constraint formulation and it encompasses the classical analytical constraint concept.

The GPME for rigid body models subjected to constraints (7) have the form:

$$\mathbf{M}(\mathbf{q}) \ddot{\mathbf{q}} - \mathbf{h}(\mathbf{q}, \dot{\mathbf{q}}) + \mathbf{g}(\mathbf{q}) = \mathbf{Q}(t, \mathbf{q}, \dot{\mathbf{q}}), \quad (8.1)$$

$$\mathbf{B}(t, \mathbf{q}, \dot{\mathbf{q}}, \dots, \mathbf{q}^{(p-1)}) \mathbf{q}^{(p)} + \mathbf{s}(t, \mathbf{q}, \dot{\mathbf{q}}, \dots, \mathbf{q}^{(p-1)}) = \mathbf{0}, \quad (8.2)$$

where: $\mathbf{M}(\mathbf{q})$ is mass matrix, $\mathbf{h}(\mathbf{q}, \dot{\mathbf{q}})$ is a vector of centrifugal forces, $\mathbf{g}(\mathbf{q})$ is a vector of gravity forces, $\mathbf{Q}(t, \mathbf{q}, \dot{\mathbf{q}})$ is a vector of generalized forces vector resulting from external actions which are not controls. The set of equations (8.1) and (8.2) can be transformed to the form that uses directly functions for the kinetic and potential energies, as well as explicit forms of external forces on a system.

The GPME modified for flexible open-loop structures subjected to position constraints are detailed in [39, 40].

For the development of a general structure of the automated CoPCoD, we take advantage of the form of the GPME that uses the energy and force functions. Also, the actual form of the CoPCoD can incorporate position and velocity constraints, both material and non-material into

system dynamics. From engineering application points of view such constraints are the most significant ones. Modifications that are dedicated to motion description of rigid and flexible open and closed-loop substructures have to be substantiated. The modified new structure of the GPME for position and velocity constraints is introduced as follows

$$\frac{\partial R_l}{\partial \dot{q}_v} + \sum_{w \in i_{d_c}} \frac{\partial R_l}{\partial \dot{q}_w} \frac{\partial \dot{q}_w}{\partial \dot{q}_v} = 0, \quad v \in i_c, \quad (9.1)$$

$$\Phi(t, \mathbf{q}) = \mathbf{0}, \quad (9.2)$$

where now the function R_l that collects the system energies and forces is composed of

$$R_l = \begin{cases} R_l^{(b)} + R_l^{(c)} & \text{for OLKC,} \\ R_l^{(b)} + \sum_{\beta \in \{c_1, c_2\}} R_l^{(\beta)} & \text{for CLKC,} \end{cases}$$

$$R_l^{(b)} = \dot{E}_k^{(b)} - 2 \sum_{v=1}^{n_{dof}^{(b)}} \frac{\partial E_k^{(b)}}{\partial \dot{q}_v^{(b)}} \dot{q}_v^{(b)} + \sum_{v=1}^{n_{dof}^{(b)}} \frac{\partial E_p^{(b)}}{\partial \dot{q}_v^{(b)}} \dot{q}_v^{(b)} + \sum_{v=1}^{n_{dof}^{(b)}} \frac{\partial R_{sup}^{(b)}}{\partial \dot{q}_v^{(b)}} \dot{q}_v^{(b)},$$

$$R_l^{(\alpha)} = E_k^{(\alpha)} - 2 \sum_{v=1}^{n_{dof}^{(\alpha)}} \frac{\partial E_k^{(\alpha)}}{\partial \dot{q}_v^{(\alpha)}} \dot{q}_v^{(\alpha)} + \sum_{v=1}^{n_{dof}^{(\alpha)}} \frac{\partial E_p^{(\alpha)}}{\partial \dot{q}_v^{(\alpha)}} \dot{q}_v^{(\alpha)} + \sum_{v=1}^{n_{dof}^{(\alpha)}} \frac{\partial R_{f_l}^{(\alpha)}}{\partial \dot{q}_v^{(\alpha)}} \dot{q}_v^{(\alpha)} + \sum_{v=1}^{n_{dof}^{(\alpha)}} f_{fric,v}^{(\alpha)} \dot{q}_v^{(\alpha)},$$

$E_k^{(b)}, E_k^{(\alpha)} \Big|_{\alpha \in \{c, c_1, c_2\}}$ are kinetic energy terms of the flexibly supported base and the chains,

$E_p^{(b)} = E_{p,sup}^{(b)} + E_{p,g}^{(b)}$ is the potential energy of the base which can be calculated as a sum of spring deformations energy and the potential energy of the gravity forces,

$E_p^{(\alpha)} \Big|_{\alpha \in \{c, c_1, c_2\}} = E_{p,g}^{(\alpha)} + E_{p,f_l}^{(\alpha)}$ is the potential energy of the chain which can be expressed as a sum of spring deformations energy of flexible links and the potential energy of the gravity forces of links,

$R_{sup}^{(b)}, R_{f_l}^{(\alpha)} \Big|_{\alpha \in \{c, c_1, c_2\}}$ are the Rayleigh dissipation functions of the flexibly supported base and the chains,

$\mathbf{f}_{fric}^{(\alpha)} \Big|_{\alpha \in \{c, c_1, c_2\}}$ is the vector of friction forces acting in the joints,

$\Phi(t, \mathbf{q})$ are the programmed and kinematic constraint equations.

Thus, in order to derive the CoPCoD, it is required to determine the kinetic and potential energies, the Rayleigh dissipation functions and external forces for components of kinematic chains, as well as formulate the constraint equations.

2.2.1. Kinetic energy of the system

The kinetic energy of the system structure can be presented in the following forms:

$$E_k^{(b)} = \frac{1}{2} \text{tr} \left(\dot{\mathbf{T}}^{(b)} \mathbf{H}^{(b)} \dot{\mathbf{T}}^{(b)T} \right), \quad (10.1)$$

$$E_k^{(\alpha)} \Big|_{\alpha \in \{c, c_1, c_2\}} = \frac{1}{2} \sum_{i \in I_{\eta}^{(\alpha)}} \text{tr} \left(\dot{\mathbf{T}}^{(\alpha,i)} \mathbf{H}^{(\alpha,i)} \dot{\mathbf{T}}^{(\alpha,i)T} \right) + \frac{1}{2} \sum_{i \in I_{f_l}^{(\alpha)}} \sum_{r=0}^{n_{rfe}^{(\alpha,i)}-1} \text{tr} \left(\dot{\mathbf{T}}^{(\alpha,i,r)} \mathbf{H}^{(\alpha,i,r)} \dot{\mathbf{T}}^{(\alpha,i,r)T} \right), \quad (10.2)$$

where:

$\mathbf{H}^{(\gamma)} \Big|_{\gamma \in \{b, (\alpha, i), (\alpha, i, r)\}}$ is the pseudo-inertia matrix of the link, $\mathbf{H}^{(\gamma)} = \begin{bmatrix} h_{(yz)}^{(\gamma)} & h_{xy}^{(\gamma)} & h_{xz}^{(\gamma)} & h_x^{(\gamma)} \\ & h_{(xz)}^{(\gamma)} & h_{yz}^{(\gamma)} & h_y^{(\gamma)} \\ & & h_{(xy)}^{(\gamma)} & h_z^{(\gamma)} \\ \text{sym.} & & & m^{(\gamma)} \end{bmatrix}$,

$m^{(\gamma)}$ is the mass of the link, $h_{\xi}^{(\gamma)} \Big|_{\xi \in \{x, y, z\}}$ are the static moments of inertia, $h_{\xi}^{(\gamma)} \Big|_{\xi \in \{xy, xz, yz\}}$ are the products of inertia, $h_{\xi}^{(\gamma)} \Big|_{\xi \in \{(yz), (xz), (xy)\}}$ are the moments of inertia with respect to the planes of the local reference frame of the link, $I_{r_l}^{(a)}, I_{f_l}^{(a)}$ are the indices of rigid and flexible links.

In Eqs (9.1) derivatives of the function R_l components with respect to time and generalized coordinates require determining the following derivatives:

$$\dot{E}_k^{(b)} = \text{tr} \left(\dot{\mathbf{T}}^{(b)} \mathbf{H}^{(b)} \dot{\mathbf{T}}^{(b)T} \right), \quad (11.1)$$

$$\dot{E}_k^{(\alpha)} \Big|_{\alpha \in \{c, c_1, c_2\}} = \sum_{i \in I_{r_l}^{(\alpha)}} \text{tr} \left(\dot{\mathbf{T}}^{(\alpha, i)} \mathbf{H}^{(\alpha, i)} \dot{\mathbf{T}}^{(\alpha, i)T} \right) + \sum_{i \in I_{f_l}^{(\alpha)}} \sum_{r=0}^{n_{rfe}^{(\alpha, i)} - 1} \text{tr} \left(\dot{\mathbf{T}}^{(\alpha, i, r)} \mathbf{H}^{(\alpha, i, r)} \dot{\mathbf{T}}^{(\alpha, i, r)T} \right), \quad (11.2)$$

$$\frac{\partial E_k^{(b)}}{\partial q_v^{(b)}} = \text{tr} \left(\mathbf{T}_v^{(b)} \mathbf{H}^{(b)} \dot{\mathbf{T}}^{(b)T} \right), \quad (11.3)$$

$$\frac{\partial E_k^{(\alpha)}}{\partial q_v^{(\alpha)}} \Big|_{\alpha \in \{c, c_1, c_2\}} = \sum_{i \in I_{r_l}^{(\alpha)}} \text{tr} \left(\mathbf{T}_v^{(\alpha, i)} \mathbf{H}^{(\alpha, i)} \dot{\mathbf{T}}^{(\alpha, i)T} \right) + \sum_{i \in I_{f_l}^{(\alpha)}} \sum_{r=0}^{n_{rfe}^{(\alpha, i)} - 1} \text{tr} \left(\mathbf{T}_v^{(\alpha, i, r)} \mathbf{H}^{(\alpha, i, r)} \dot{\mathbf{T}}^{(\alpha, i, r)T} \right), \quad (11.4)$$

where $\mathbf{T}_v^{(\gamma)} \Big|_{\gamma \in \{b, (\alpha, i), (\alpha, i, r)\}} = \frac{\partial \mathbf{T}^{(\gamma)}}{\partial q_v^{(\gamma)}}.$

The time derivatives of the transformation matrices are determined by the following formulas:

$$\dot{\mathbf{T}}^{(\gamma)} \Big|_{\gamma \in \{b, (\alpha, i), (\alpha, i, r)\}} = \sum_{v=1}^{n_{dof}^{(\gamma)}} \mathbf{T}_v^{(\gamma)} \dot{q}_v^{(\gamma)}, \quad (12.1)$$

$$\ddot{\mathbf{T}}^{(\gamma)} \Big|_{\gamma \in \{b, (\alpha, i), (\alpha, i, r)\}} = \sum_{v=1}^{n_{dof}^{(\gamma)}} \mathbf{T}_v^{(\gamma)} \ddot{q}_v^{(\gamma)} + \sum_{v=1}^{n_{dof}^{(\gamma)}} \sum_{w=1}^{n_{dof}^{(\gamma)}} \mathbf{T}_{vw}^{(\gamma)} \dot{q}_v^{(\gamma)} \dot{q}_w^{(\gamma)}, \quad (12.2)$$

where: $\mathbf{T}_{v,w}^{(\gamma)} \Big|_{\gamma \in \{b, (\alpha, i), (\alpha, i, r)\}} = \frac{\partial^2 \mathbf{T}^{(\gamma)}}{\partial q_v^{(\gamma)} \partial q_w^{(\gamma)}}.$

2.2.2. Potential energy of the system gravity force

The potential energy of the gravity force that acts on the chains can be presented as follows:

$$E_{p,g}^{(b)} = m^{(b)} g \mathbf{j}_2 \mathbf{T}^{(b)} \tilde{\mathbf{r}}_C^{(b)}, \quad (13.1)$$

$$E_{p,g}^{(a)} \Big|_{\alpha \in \{c, c_1, c_2\}} = \sum_{i \in I_{r_l}^{(a)}} m^{(\alpha, i)} g \mathbf{j}_2 \mathbf{T}^{(\alpha, i)} \tilde{\mathbf{r}}_C^{(\alpha, i)} + \sum_{i \in I_{f_l}^{(a)}} \sum_{r=0}^{n_{rfe}^{(\alpha, i)} - 1} m^{(\alpha, i, r)} g \mathbf{j}_2 \mathbf{T}^{(\alpha, i, r)} \tilde{\mathbf{r}}_C^{(\alpha, i, r)}, \quad (13.2)$$

where g is the gravity acceleration, $\mathbf{r}_C^{(\gamma)}|_{\gamma \in \{b, (\alpha, i), (\alpha, i, r)\}}$ are position vectors of the mass center of

the link, $\mathbf{j}_2$ is the second row of a reducing matrix $\mathbf{J} = \begin{bmatrix} \mathbf{j}_1 \\ \mathbf{j}_2 \\ \mathbf{j}_3 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}$.

The derivatives of Eqs (13.1) and (13.2) with respect to the generalized coordinates are written as follows:

$$\frac{\partial E_{p,g}^{(b)}}{\partial q_v^{(b)}} = m^{(b)} g \mathbf{j}_2 \mathbf{T}_v^{(b)} \tilde{\mathbf{r}}_C^{(b)}, \quad (14.1)$$

$$\frac{\partial E_{p,g}^{(\alpha)}}{\partial q_v^{(\alpha)}} = \sum_{i \in I_\eta^{(\alpha)}} m^{(\alpha, i)} g \mathbf{j}_2 \mathbf{T}_v^{(\alpha, i)} \tilde{\mathbf{r}}_C^{(\alpha, i)} + \sum_{i \in I_{\eta}^{(\alpha)}} \sum_{r=0}^{n_{\eta}^{(\alpha, i)} - 1} m^{(\alpha, i, r)} g \mathbf{j}_2 \mathbf{T}_v^{(\alpha, i, r)} \tilde{\mathbf{r}}_C^{(\alpha, i, r)}. \quad (14.2)$$

2.2.3. Potential energy of spring deformations and the Rayleigh dissipation function of the base support

The flexible support of the base is modelled by means of spring-damping elements as shown in Fig. 2. Positions of the supports are described with respect to the base fixed frame. It is assumed that deformations of the spring-damping elements are small and the transformation matrix can be linearized.

Fig. 2. Flexible support of the base

The spring deformation energy and the Rayleigh dissipation function of the spring-damping elements are determined according to the following formulas:

$$E_{p,sup}^{(b)} = \frac{1}{2} \sum_{i=1}^{n_e} \sum_{j=1}^{n_{sde_s}^{(i)}} (\mathbf{d}_{sde_s}^{(i,j)})^T \mathbf{s}_{sde_s}^{(i,j)} \mathbf{d}_{sde_s}^{(i,j)}, \quad (15.1)$$

$$R_{sup}^{(b)} = \frac{1}{2} \sum_{i=1}^{n_e} \sum_{j=1}^{n_{sde_s}^{(i)}} \left(\dot{\mathbf{d}}_{sde_s}^{(i,j)} \right)^T \mathbf{D}_{sde_s}^{(i,j)} \dot{\mathbf{d}}_{sde_s}^{(i,j)}, \quad (15.2)$$

where: $\mathbf{d}_{sde_s}^{(i,j)}$ is a vector of deformation of $sde_s(i, j)$, $\mathbf{d}_{sde_s}^{(i,j)} = \mathbf{J} \Delta \mathbf{r}_{E_{ij}}^{(b)} = \mathbf{J} \left(\mathbf{T}^{(b)} \tilde{\mathbf{r}}_{E_{ij}}^{(b)} - \mathbf{r}_{E_{ij,0}}^{(b)} \right)$, $\mathbf{r}_{E_{ij,0}}^{(b)}$ is a position vector of point E_{ij} in the undeformed state,

$\mathbf{S}_{sde_s}^{(i,j)} = \text{diag} \left(s_{sde_s,x}^{(i,j)}, s_{sde_s,y}^{(i,j)}, s_{sde_s,z}^{(i,j)} \right)$, $\mathbf{D}_{sde_s}^{(i,j)} = \text{diag} \left(d_{sde_s,x}^{(i,j)}, d_{sde_s,y}^{(i,j)}, d_{sde_s,z}^{(i,j)} \right)$ are matrices containing stiffness and damping coefficients of $sde_s(i, j)$.

The derivatives of Eqs (15.1) and (15.2) with respect to the generalized coordinates and velocities are written as follows:

$$\frac{\partial E_{p,sup}^{(b)}}{\partial \dot{q}_v^{(b)}} = \sum_{i=1}^{n_e} \sum_{j=1}^{n_{sde_s}^{(i)}} \left(\frac{\partial \mathbf{d}_{sde_s}^{(i,j)}}{\partial \dot{q}_v^{(b)}} \right)^T \mathbf{S}_{sde_s}^{(i,j)} \mathbf{d}_{sde_s}^{(i,j)}, \quad (16.1)$$

$$\frac{\partial R_{sup}^{(b)}}{\partial \dot{q}_v^{(b)}} = \sum_{i=1}^{n_e} \sum_{j=1}^{n_{sde_s}^{(i)}} \left(\frac{\partial \dot{\mathbf{d}}_{sde_s}^{(i,j)}}{\partial \dot{q}_v^{(b)}} \right)^T \mathbf{D}_{sde_s}^{(i,j)} \dot{\mathbf{d}}_{sde_s}^{(i,j)}. \quad (16.2)$$

2.2.4. Potential energy of spring deformations and the Rayleigh dissipation function of flexible links

The energy of spring deformations and the Rayleigh dissipation function of flexible links are written in the following forms:

$$E_{p,f_l}^{(\alpha)} \Big|_{\alpha \in \{c, c_1, c_2\}} = \frac{1}{2} \sum_{i \in I_{f_l}^{(\alpha)}} \sum_{j=1}^{n_{sde_l}^{(\alpha,i,j)}} \left(\mathbf{d}_{sde_l}^{(\alpha,i,j)} \right)^T \mathbf{S}_{sde_l}^{(\alpha,i,j)} \mathbf{d}_{sde_l}^{(\alpha,i,j)}, \quad (17.1)$$

$$R_{f_l}^{(\alpha)} \Big|_{\alpha \in \{c, c_1, c_2\}} = \frac{1}{2} \sum_{i \in I_{f_l}^{(\alpha)}} \sum_{j=1}^{n_{sde_l}^{(\alpha,i,j)}} \left(\dot{\mathbf{d}}_{sde_l}^{(\alpha,i,j)} \right)^T \mathbf{D}_{sde_l}^{(\alpha,i,j)} \dot{\mathbf{d}}_{sde_l}^{(\alpha,i,j)}, \quad (17.2)$$

where $\mathbf{d}_{sde_l}^{(\alpha,i,j)}$ is a vector of deformation of $sde_l(\alpha, i, j)$, $\mathbf{d}_{sde_l}^{(\alpha,i,j)} = \tilde{\mathbf{q}}^{(\alpha,i,j)}$,

$\mathbf{S}_{sde_l}^{(\alpha,i,j)} = \text{diag} \left(s_{sde_l,\psi}^{(\alpha,i,j)}, s_{sde_l,\theta}^{(\alpha,i,j)}, s_{sde_l,\varphi}^{(\alpha,i,j)} \right)$, $\mathbf{D}_{sde_l}^{(\alpha,i,j)} = \text{diag} \left(d_{sde_l,\psi}^{(\alpha,i,j)}, d_{sde_l,\theta}^{(\alpha,i,j)}, d_{sde_l,\varphi}^{(\alpha,i,j)} \right)$ are stiffness and damping matrices of $sde_l(\alpha, i, j)$.

The derivatives of Eqs (17.1) and (17.2) with respect to the generalized coordinates and velocities are written as:

$$\frac{\partial E_{p,f_l}^{(\alpha)} \Big|_{\alpha \in \{c, c_1, c_2\}}}{\partial \tilde{\mathbf{q}}^{(\alpha,i,j)}} = \sum_{i \in I_{f_l}^{(\alpha)}} \sum_{j=1}^{n_{sde_l}^{(\alpha,i,j)}} \mathbf{S}_{sde_l}^{(\alpha,i,j)} \tilde{\mathbf{q}}^{(\alpha,i,j)}, \quad (18.1)$$

$$\frac{\partial R_{f_l}^{(\alpha)} \Big|_{\alpha \in \{c, c_1, c_2\}}}{\partial \dot{\tilde{\mathbf{q}}}^{(\alpha,i,j)}} = \sum_{i \in I_{f_l}^{(\alpha)}} \sum_{j=1}^{n_{sde_l}^{(\alpha,i,j)}} \mathbf{D}_{sde_l}^{(\alpha,i,j)} \dot{\tilde{\mathbf{q}}}^{(\alpha,i,j)}. \quad (18.2)$$

It can be seen that deformations and deformation velocities of $\text{sde}_l(\alpha, i, j)$ are taken directly from the generalized coordinates vector and its derivative.

2.2.5. Generalized forces resulting from actions of friction torques and forces

The LuGre friction model [48] is used to describe friction phenomenon in joints. Planar models of revolute and prismatic joints with friction are presented in Fig. 3.

Fig. 3. Friction models in revolute (a) and prismatic (b) joints.

The generalized forces vector of a chain α resulting from including friction in joints can be written as follows:

$$\mathbf{f}_{fric}^{(\alpha)} \Big|_{\alpha \in \{c, c_1, c_2\}} = \sum_{i \in I_{\eta}^{(\alpha)}} \mathbf{f}_{fric}^{(\alpha, i)} + \sum_{i \in I_{f_l}^{(\alpha)}} \mathbf{f}_{fric}^{(\alpha, i, 0)}, \quad (19)$$

where $\mathbf{f}_{fric}^{(\alpha, i)} = [\mathbf{0} \quad \dots \quad f_{fric, j}^{(\alpha, i)} \quad \dots \quad \mathbf{0}]$.

Forces $f_{fric, j}^{(\alpha, i)}$ are non-zero elements containing friction forces acting on links (α, i) and can be presented in the form:

$$f_{fric, j}^{(\alpha, i)} \Big|_{\alpha \in \{c, c_1, c_2\}} = \begin{cases} t_f^{(\alpha, i)} & \text{for a revolute joint,} \\ f_f^{(\alpha, i)} & \text{for a prismatic joint.} \end{cases} \quad (20)$$

Friction torques and forces are calculated as follows:

$$t_f^{(\alpha, i)} = \mu^{(\alpha, i)} \tilde{f}_n^{(\alpha, i)} \frac{d^{(\alpha, i)}}{2}, \quad (21.1)$$

$$f_f^{(\alpha, i)} = \mu^{(\alpha, i)} \tilde{f}_n^{(\alpha, i)}, \quad (21.2)$$

where: $\mu^{(\alpha, i)}$ is a friction coefficient,

$$\tilde{f}_n^{(\alpha, i)} \text{ is a normal force in a joint, } \tilde{f}_n^{(\alpha, i)} = \begin{cases} \sqrt{(\tilde{f}_x^{(\alpha, i)})^2 + (\tilde{f}_y^{(\alpha, i)})^2} & \text{for a revolute joint,} \\ \tilde{f}_x^{(\alpha, i)} & \text{for a prismatic joint.} \end{cases}$$

$\tilde{f}_x^{(\alpha, i)}, \tilde{f}_y^{(\alpha, i)}$ are joint forces, $d^{(\alpha, i)}$ is a diameter of the journal.

The joint forces can be determined, in each integration step of the dynamics equations of motion, using the recursive Newton-Euler algorithm [49].

Friction coefficients are calculated using the LuGre friction model from the following formula:

$$\mu^{(a,i)} = \sigma_0^{(a,i)} z^{(a,i)} + \sigma_1^{(a,i)} \dot{z}^{(a,i)} + \sigma_2^{(a,i)} \ddot{q}^{(a,i)}, \quad (22)$$

where: $\sigma_0^{(a,i)}, \sigma_1^{(a,i)}, \sigma_2^{(a,i)}$ are stiffness, damping and viscous friction coefficients, $z^{(a,i)}$ is the state

$$\text{variable, } \dot{z}^{(a,i)} = \dot{\ddot{q}}^{(a,i)} \left(1 - \frac{\sigma_0^{(a,i)} z^{(a,i)} \operatorname{sgn}(\ddot{q}^{(a,i)})}{\mu_k^{(a,i)} + (\mu_s^{(a,i)} - \mu_k^{(a,i)}) \exp\left(-\left(\frac{\ddot{q}^{(a,i)}}{\dot{q}_{\text{Stribeck}}^{(a,i)}}\right)^2\right)} \right),$$

$\mu_s^{(a,i)}, \mu_k^{(a,i)}$ are static and kinetic friction coefficients, $\dot{q}_{\text{Stribeck}}^{(a,i)}$ is the Stribeck velocity.

2.3. Kinematic and programmed constraint equations

The GPME (9.1) needs to be supplemented by constraint equations composed of n_k kinematic and n_p programmed constraint equations which can be written in the following form:

$$\Phi(t, \mathbf{q}) \equiv \begin{cases} \Phi_p(\mathbf{q}) + \mathbf{g}_p(t) = \mathbf{0} & \text{for OLKC,} \\ \begin{cases} \Phi_k(\mathbf{q}) = \mathbf{0} \\ \Phi_p(\mathbf{q}) + \mathbf{g}_p(t) = \mathbf{0} \end{cases} & \text{for CLKC,} \end{cases} \quad (23)$$

where $\Phi_k(\mathbf{q}), \Phi_p(\mathbf{q})$ are kinematic and programmed constraints equations, $\mathbf{g}_p(t)$ is a vector of known time functions. The kinematic constraints come from closed chains modelling and are material constraints.

Time derivatives of the constraint equations (23) lead to:

$$\dot{\Phi}(t, \mathbf{q}, \dot{\mathbf{q}}) \equiv \begin{cases} \Phi_{p,q}(\mathbf{q}) \dot{\mathbf{q}} + \dot{\mathbf{g}}_p(t) = \mathbf{0} & \text{for OLKC,} \\ \begin{cases} \Phi_{k,q}(\mathbf{q}) \dot{\mathbf{q}} = \mathbf{0} \\ \Phi_{p,q}(\mathbf{q}) \dot{\mathbf{q}} + \dot{\mathbf{g}}_p(t) = \mathbf{0} \end{cases} & \text{for CLKC,} \end{cases} \quad (24.1)$$

$$\ddot{\Phi}(t, \mathbf{q}, \dot{\mathbf{q}}, \ddot{\mathbf{q}}) \equiv \begin{cases} \Phi_{p,q}(\mathbf{q}) \ddot{\mathbf{q}} + \dot{\Phi}_{p,q}(\mathbf{q}) \dot{\mathbf{q}} + \ddot{\mathbf{g}}_p(t) = \mathbf{0} & \text{for OLKC,} \\ \begin{cases} \Phi_{k,q}(\mathbf{q}) \ddot{\mathbf{q}} + \dot{\Phi}_{k,q}(\mathbf{q}) \dot{\mathbf{q}} = \mathbf{0} \\ \Phi_{p,q}(\mathbf{q}) \ddot{\mathbf{q}} + \dot{\Phi}_{p,q}(\mathbf{q}) \dot{\mathbf{q}} + \ddot{\mathbf{g}}_p(t) = \mathbf{0} \end{cases} & \text{for CLKC,} \end{cases} \quad (24.2)$$

where: $\Phi_{k,q}(\mathbf{q}) = \frac{\partial \Phi_k}{\partial \mathbf{q}}, \Phi_{p,q}(\mathbf{q}) = \frac{\partial \Phi_p}{\partial \mathbf{q}}$ are kinematic and programmed constraint matrices.

The vector of depended velocities calculated from Eqs. (24.1) takes the form:

$$\dot{\mathbf{q}}_{d_c} = \Phi_{q,d_c}^{-1} \Phi_{q,i_c} \dot{\mathbf{q}}_{i_c}, \quad (25)$$

where: $\Phi_{q,i_c}, \Phi_{q,d_c}$ are obtained the from the global constraint matrix $\Phi_q = \begin{bmatrix} \Phi_{k,q} \\ \Phi_{p,q} \end{bmatrix}$ by selecting

rows and columns corresponding to indices of the independent and dependent variables in vector $\mathbf{q}$.

2.4. The computational procedure for constrained dynamics (CoPCoD) for systems subjected to constraints – the general form

Substituting components from Eqs (11.1) – (11.4), (14.1), (14.2), (16.1), (16.2), (18.1), (18.2), (19), (24.1) and (24.2), to the GMPE Eqs (9.1), and performing all transformations, the CoPCoD underlying equations are obtained in the following matrix form:

$$\begin{bmatrix} \mathbf{M}_{i_c} + \mathbf{M}_{d_c} \\ \Phi_{p,q} \\ \bar{\Phi}_{k,q} \end{bmatrix} \ddot{\mathbf{q}} = \begin{bmatrix} \mathbf{h}_{i_c} + \mathbf{Q}_{i_c} + \Delta \mathbf{Q}_{i_c} + \mathbf{h}_{d_c} + \mathbf{Q}_{d_c} + \Delta \mathbf{Q}_{d_c} \\ -\dot{\Phi}_{p,q} \dot{\mathbf{q}} - \ddot{\mathbf{g}}_p(t) \\ -\dot{\Phi}_{k,q} \dot{\mathbf{q}} \end{bmatrix}, \quad (26)$$

where:

$$\mathbf{M}_{i_c} = (\text{row}_i(\mathbf{M}))_{i \in i_c}, \quad \mathbf{M}_{d_c} = \left(\sum_{j \in i_{d_c}} \text{row}_j(\mathbf{M}) \frac{\partial \dot{q}_j}{\partial \dot{q}_i} \right)_{i \in i_c}, \quad \text{row}_i(\cdot) \text{ is the } i\text{-th matrix row},$$

$$\mathbf{M} = \begin{cases} \mathbf{M}^{(b)} + \mathbf{M}^{(c)} & \text{for OLKC,} \\ \mathbf{M}^{(b)} + \sum_{\beta \in \{c_1, c_2\}} \mathbf{M}^{(\beta)} & \text{for CLKC,} \end{cases}$$

$$\mathbf{M}^{(b)} = \begin{bmatrix} \mathbf{M}_{1,1}^{(b)} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix}, \quad \mathbf{M}_{11}^{(b)} = (m_{i,j}^{(b)})_{i,j=1,\dots,n_{dof}^{(b)}}, \quad m_{i,j}^{(b)} = \text{tr} \left\{ \mathbf{T}_i^{(b)} \mathbf{H}^{(b)} \mathbf{T}_j^{(b)^T} \right\},$$

$$\mathbf{M}^{(c)} = \begin{bmatrix} \mathbf{M}_{1,1}^{(c)} & \dots & \mathbf{M}_{1,j}^{(c)} & \dots & \mathbf{M}_{1,n_l^{(c)}+1}^{(c)} \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ \mathbf{M}_{i,1}^{(c)} & \dots & \mathbf{M}_{i,j}^{(c)} & \dots & \mathbf{M}_{i,n_l^{(c)}+1}^{(c)} \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ \mathbf{M}_{n_l^{(c)}+1,1}^{(c)} & \dots & \mathbf{M}_{n_l^{(c)}+1,j}^{(c)} & \dots & \mathbf{M}_{n_l^{(c)}+1,n_l^{(c)}+1}^{(c)} \end{bmatrix},$$

$$\mathbf{M}^{(c_1)} = \begin{bmatrix} \mathbf{M}_{1,1}^{(c_1)} & \dots & \mathbf{M}_{1,j}^{(c_1)} & \dots & \mathbf{M}_{1,n_l^{(c_1)}+1}^{(c_1)} & \mathbf{0} \\ \vdots & \ddots & \vdots & \ddots & \vdots & \vdots \\ \mathbf{M}_{i,1}^{(c_1)} & \dots & \mathbf{M}_{i,j}^{(c_1)} & \dots & \mathbf{M}_{i,n_l^{(c_1)}+1}^{(c_1)} & \mathbf{0} \\ \vdots & \ddots & \vdots & \ddots & \vdots & \vdots \\ \mathbf{M}_{n_l^{(c_1)}+1,1}^{(c_1)} & \dots & \mathbf{M}_{n_l^{(c_1)}+1,j}^{(c_1)} & \dots & \mathbf{M}_{n_l^{(c_1)}+1,n_l^{(c_1)}+1}^{(c_1)} & \mathbf{0} \\ \mathbf{0} & \dots & \mathbf{0} & \dots & \mathbf{0} & \mathbf{0} \end{bmatrix},$$

$$\mathbf{M}^{(c_2)} = \begin{bmatrix} \mathbf{M}_{1,1}^{(c_2)} & \mathbf{0} & \mathbf{M}_{1,2}^{(c_2)} & \dots & \mathbf{M}_{1,j}^{(c_2)} & \dots & \mathbf{M}_{1,n_l}^{(c_2)} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \dots & \mathbf{0} & \dots & \mathbf{0} \\ \mathbf{M}_{2,1}^{(c_2)} & \mathbf{0} & \mathbf{M}_{2,2}^{(c_2)} & \dots & \mathbf{M}_{2,j}^{(c_2)} & \dots & \mathbf{M}_{2,n_l}^{(c_2)} \\ \vdots & \vdots & \vdots & \ddots & \vdots & \ddots & \vdots \\ \mathbf{M}_{i,1}^{(c_2)} & \dots & \mathbf{M}_{i,2}^{(c_2)} & \dots & \mathbf{M}_{i,j}^{(c_2)} & \dots & \mathbf{M}_{i,n_l}^{(c_2)} \\ \vdots & \vdots & \vdots & \ddots & \vdots & \ddots & \vdots \\ \mathbf{M}_{n_l,1}^{(c_2)} & \dots & \mathbf{M}_{n_l,2}^{(c_2)} & \dots & \mathbf{M}_{n_l,j}^{(c_2)} & \dots & \mathbf{M}_{n_l,n_l}^{(c_2)} \end{bmatrix},$$

$$\mathbf{M}_{i,j}^{(\beta)} = \sum_{l=\max\{i,j\}}^{n_l^{(\beta)}} \mathbf{M}_{i,j}^{(\beta,l)}, \mathbf{M}_{i,j}^{(\beta,l)} \Big|_{i,j=1,\dots,l} = \left(m_{n_{dof}^{(\beta,j-1)}+v, n_{dof}^{(\beta,j-1)}+w}^{(\beta,l)} \right)_{v=1,\dots,\tilde{n}_{dof}^{(\beta,j)}, w=1,\dots,\tilde{n}_{dof}^{(\beta,j)}}, m_{i,j}^{(\beta,l)} = \text{tr} \left\{ \mathbf{T}_i^{(\beta,l)} \mathbf{H}^{(\beta,l)} \mathbf{T}_j^{(\beta,l)T} \right\},$$

$$\mathbf{h}_{i_c} = (h_i)_{i \in i_c}, \mathbf{h}_{d_c} = \left(\sum_{j \in i_{d_c}} h_j \frac{\partial \dot{q}_j}{\partial \dot{q}_i} \right)_{i \in i_c},$$

$$\mathbf{h} = \begin{cases} \mathbf{h}^{(b)} + \mathbf{h}^{(c)} & \text{for OLKC,} \\ \mathbf{h}^{(b)} + \sum_{\beta \in \{c_1, c_2\}} \mathbf{h}^{(\beta)} & \text{for CLKC,} \end{cases}$$

$$\mathbf{h}^{(b)} = \begin{bmatrix} \mathbf{h}_1^{(b)} \\ \mathbf{0} \end{bmatrix}, \mathbf{h}_1^{(b)} = (h_i^{(b)})_{i,j=1,\dots,n_{dof}^{(b)}},$$

$$h_i^{(b)} = \sum_{m=1}^{n_{dof}^{(b)}} \sum_{n=1}^{n_{dof}^{(b)}} \text{tr} \left\{ \mathbf{T}_m^{(b)} \mathbf{H}^{(b)} (\mathbf{T}_{m,n}^{(b)})^T \right\} \dot{q}_m^{(b)} \dot{q}_n^{(b)} + 2 \sum_{m=1}^{n_{dof}^{(b)}} \sum_{n=1}^{n_{dof}^{(b)}} \text{tr} \left\{ \mathbf{T}_m^{(b)} \mathbf{H}^{(b)} (\mathbf{T}_{i,n}^{(b)})^T \right\} \dot{q}_m^{(b)} \dot{q}_n^{(b)},$$

$$\mathbf{h}^{(c)} = \begin{bmatrix} \mathbf{h}_1^{(c)} \\ \vdots \\ \mathbf{h}_i^{(c)} \\ \vdots \\ \mathbf{h}_{n_l^{(c)}+1}^{(c)} \end{bmatrix}, \quad \mathbf{h}^{(c_1)} = \begin{bmatrix} \mathbf{h}_1^{(c_1)} \\ \vdots \\ \mathbf{h}_i^{(c_1)} \\ \vdots \\ \mathbf{h}_{n_l^{(c_1)}+1}^{(c)} \\ \mathbf{0} \end{bmatrix}, \quad \mathbf{h}^{(c_2)} = \begin{bmatrix} \mathbf{h}_1^{(c_2)} \\ \mathbf{0} \\ \mathbf{h}_2^{(c_2)} \\ \vdots \\ \mathbf{h}_i^{(c_2)} \\ \vdots \\ \mathbf{h}_{n_l^{(c_2)}}^{(c_2)} \end{bmatrix}, \quad \mathbf{h}_i^{(\beta)} = \sum_{l=i}^{n_l^{(\beta)}} \mathbf{h}_i^{(\beta,l)},$$

$$\mathbf{h}_i^{(\beta,l)} \Big|_{i=1,\dots,l} = \left(h_{n_{dof}^{(\beta,l-1)}+v}^{(\beta,l)} \right)_{v=1,\dots,\tilde{n}_{dof}^{(\beta,l)}},$$

$$h_i^{(\beta,l)} = \sum_{m=1}^{n_{dof}^{(\beta,l)}} \sum_{n=1}^{n_{dof}^{(\beta,l)}} \text{tr} \left\{ \mathbf{T}_m^{(\beta,l)} \mathbf{H}^{(\beta,l)} (\mathbf{T}_{m,n}^{(\beta,l)})^T \right\} \dot{q}_m^{(\beta,l)} \dot{q}_n^{(\beta,l)} + 2 \sum_{m=1}^{n_{dof}^{(\beta,l)}} \sum_{n=1}^{n_{dof}^{(\beta,l)}} \text{tr} \left\{ \mathbf{T}_m^{(\beta,l)} \mathbf{H}^{(\beta,l)} (\mathbf{T}_{i,n}^{(\beta,l)})^T \right\} \dot{q}_m^{(\beta,l)} \dot{q}_n^{(\beta,l)},$$

$$\mathbf{Q}_{i_c} = (Q_i)_{i \in i_c}, \mathbf{Q}_{d_c} = \left(\sum_{j \in i_{d_c}} Q_j \frac{\partial \dot{q}_j}{\partial \dot{q}_i} \right)_{i \in i_c},$$

$$\Delta \mathbf{Q}_{i_c} = \left(\sum_{k \in i_c \cup i_{d_c}} \dot{q}_k \frac{\partial Q_k}{\partial \dot{q}_i} \right)_{i \in i_c}, \Delta \mathbf{Q}_{d_c} = \left(\sum_{j \in i_{d_c}} \sum_{k \in i_c \cup i_{d_c}} \dot{q}_k \frac{\partial Q_k}{\partial \dot{q}_i} \frac{\partial \dot{q}_j}{\partial \dot{q}_i} \right)_{i \in i_c},$$

$$\mathbf{Q} = -(\mathbf{g} + \mathbf{f}_{fl} + \mathbf{f}_{sup} + \mathbf{f}_{fric}),$$

$$\mathbf{g} = \begin{cases} \mathbf{g}^{(b)} + \mathbf{g}^{(c)} & \text{for OLKC,} \\ \mathbf{g}^{(b)} + \sum_{\beta \in \{c_1, c_2\}} \mathbf{g}^{(\beta)} & \text{for CLKC,} \end{cases}$$

$$\mathbf{g}^{(b)} = \begin{bmatrix} \mathbf{g}_1^{(b)} \\ \mathbf{0} \end{bmatrix}, \mathbf{g}_1^{(b)} = (g_i^{(b)})_{i,j=1,\dots,n_{dof}^{(b)}}, g_i^{(b)} = m^{(b)} g \mathbf{j}_2 \mathbf{T}_i^{(b)} \mathbf{r}_C^{(b)},$$

$$\mathbf{g}^{(c)} = \begin{bmatrix} \mathbf{g}_1^{(c)} \\ \vdots \\ \mathbf{g}_i^{(c)} \\ \vdots \\ \mathbf{g}_{n_l^{(c)}+1}^{(c)} \end{bmatrix}, \mathbf{g}^{(c_1)} = \begin{bmatrix} \mathbf{g}_1^{(c_1)} \\ \vdots \\ \mathbf{g}_i^{(c_1)} \\ \vdots \\ \mathbf{g}_{n_l^{(c_1)}+1}^{(c_1)} \\ \mathbf{0} \end{bmatrix}, \mathbf{g}^{(c_2)} = \begin{bmatrix} \mathbf{g}_1^{(c_2)} \\ \mathbf{0} \\ \mathbf{g}_2^{(c_2)} \\ \vdots \\ \mathbf{g}_i^{(c_2)} \\ \vdots \\ \mathbf{g}_{n_l^{(c_2)}}^{(c_2)} \end{bmatrix}, \mathbf{g}_i^{(\beta)} = \sum_{l=i}^{n_l^{(\beta)}} \mathbf{g}_i^{(\beta,l)}, \mathbf{g}_i^{(\beta,l)} \Big|_{i=1,\dots,l} = \left(\mathbf{g}_{n_{dof}^{(\beta,i-1)}+v}^{(\beta,l)} \right)_{v=1,\dots,\tilde{n}_{dof}^{(\beta,l)}},$$

$$\mathbf{g}_i^{(\beta,l)} = m^{(\beta,l)} \mathbf{g} \mathbf{j}_2 \mathbf{T}_i^{(\beta,l)} \mathbf{r}_C^{(\beta,l)},$$

$\mathbf{f}_{fl}$ is a vector of generalized forces resulting from deformation of a flexible link,

$\mathbf{f}_{sup}$ is a vector of generalized forces resulting from deformation of the support,

$\mathbf{f}_{fric}$ is a vector of friction torques/forces

$$\bar{\Phi}_{k,q} = \begin{cases} \mathbf{0} & \text{for OLKC,} \\ \Phi_{k,q} & \text{for CLKC.} \end{cases}$$

Eqs (26) form a set of n_{dof} differential algebraic-equations. It can be seen that the presented approach allows to obtain the dynamics free of the constraint reaction forces. Also, CoPCoD has been designed such that variety of system structures can be modelled by selecting options provided its block architecture.

3. Examples of application of CoPCoD applications

In this section, two sample systems representing OLKC and CLKC structures are analysed. We demonstrate how the CoPCoD algorithm enables performing efficient numerical simulations. They provide information about the system model behaviour when the programmed constraints are imposed upon motion. It enables, in turn, constrained motion planning.

3.1. The OLKC manipulator model

As the first example, the programmed motion of the spatial OLKC manipulator model is analyzed. The manipulator possesses 5 degrees of freedom with respect to the base frame $\{b\}$ (Fig. 4). It is assumed that link (1,2) can be flexible and flexibility of the ground is modelled

by means of the n_{sde_s} spring-damping elements, where $n_{sde_s} = \sum_{i=1}^{n_e} n_{sde_s}^{(i)}$.

The local frames assigned to each link and generalized coordinates of the manipulator are presented in Fig. 5. The generalized coordinates vector consists of the following

$$\mathbf{q} = \begin{bmatrix} \mathbf{q}^{(b)^T} & \underbrace{\varphi^{(1,1)} \quad \psi^{(1,2)} \quad \mathbf{q}_f^{(1,2)^T} \quad \psi^{(1,3)} \quad x^{(1,4)} \quad x^{(1,5)}}_{\mathbf{q}^{(1)^T}} \end{bmatrix}^T \quad (27)$$

Fig. 4. The OLKC manipulator model and the programmed trajectory of its end-effector E.

Fig. 5. Local frames assigned to the particular links of the manipulator

where: $\mathbf{q}^{(b)} = [x^{(b)} \ y^{(b)} \ z^{(b)} \ \psi^{(b)} \ \theta^{(b)} \ \phi^{(b)}]^T$ contains generalized coordinates of the base,

$\mathbf{q}_f^{(1,2)}$ is the vector containing generalized coordinates of rfs of the link (1, 2),

$$\mathbf{q}_f^{(1,2)} = \begin{cases} \emptyset & n_{sde}^{(1,2)} = 0, \\ \left[\underbrace{\psi^{(1,2,1)} \quad \theta^{(1,2,1)} \quad \phi^{(1,2,1)}}_{\tilde{\mathbf{q}}^{(1,2,1)T}} \quad \dots \quad \underbrace{\psi^{(1,2,n_{sde}^{(1,2)})} \quad \theta^{(1,2,n_{sde}^{(1,2)})} \quad \phi^{(1,2,n_{sde}^{(1,2)})}}_{\tilde{\mathbf{q}}^{(1,2,n_{sde}^{(1,2)})T}} \right]^T & n_{sde}^{(1,2)} > 0. \end{cases}$$

It is assumed that the end-effector E has to track the rectangle trajectory defined in $\mathbf{y}^{(P)}\mathbf{z}^{(P)}$ plane (Fig. 4) and it moves between adjacent corners of the rectangle within the scheduled time $\Delta t^{(E)} = 1$ s. The change of the end-effector coordinates in this motion is described by a 5th order polynomial:

$$s_E(t) = \frac{10\Delta s_E}{\Delta t_E^3} (t - t_{E,s})^3 - \frac{15\Delta s_E}{\Delta t_E^4} (t - t_{E,s})^4 + \frac{6\Delta s_E}{\Delta t_E^5} (t - t_{E,s})^5, \quad (28)$$

where: $\Delta s_E = s_{E,f} - s_{E,s}$, $\Delta t_E = t_{E,f} - t_{E,s}$, $s_{E,f}$, $s_{E,s}$ are positions of the end-effector at start $t_{E,s}$ and final time $t_{E,f}$ of its motion. According to the motion description specified for E , the programmed constraint equations have the following form:

$$\Phi_p(t, \mathbf{q}) \equiv \mathbf{J}(\mathbf{r}_E - \hat{\mathbf{r}}_E(t)) = \mathbf{0}, \quad (29)$$

where: $\mathbf{r}_E = \mathbf{B}^{(1,5)} \tilde{\mathbf{r}}_E^{(1,5)}$ is the calculated global position of the end-effector E at time t ,

$\tilde{\mathbf{r}}_E^{(1,5)}$ is the position of the end-effector defined in the local frame of the link (1, 5),

$\hat{\mathbf{r}}_E(t) = \mathbf{B}^{(P)} \tilde{\mathbf{r}}_E^{(P)}(t)$ is the required position of the end-effector expressed in the global reference frame,

$\mathbf{B}^{(P)} = \begin{bmatrix} \mathbf{R}^{(P)} & \mathbf{r}_{org}^{(P)} \\ \mathbf{0} & 1 \end{bmatrix}$ is the constant transformation matrix from the frame $\{P\}$ to the global frame,

$\mathbf{R}^{(P)}$ is the rotation matrix from $\{P\}$ to $\{0\}$,

$\mathbf{r}_{org}^{(P)}$ is the position vector of the frame $\{P\}$ origin with respect to the global coordinate system,

$\tilde{\mathbf{r}}_E^{(P)}(t)$ is the required position of the end-effector E expressed directly in the frame $\{P\}$.

By differentiating (29) with respect to time, one obtains the constraint equations at the velocity and acceleration levels:

$$\dot{\Phi}_p(t, \mathbf{q}, \dot{\mathbf{q}}) \equiv \mathbf{J} \left(\sum_{i=1}^{n_{dof}^{(1,5)}} \mathbf{B}_i^{(1,5)} \tilde{\mathbf{r}}_E^{(1,5)} \dot{\mathbf{q}}_i^{(1,5)} - \mathbf{B}^{(P)} \dot{\hat{\mathbf{r}}}_E^{(P)}(t) \right) = \mathbf{0}, \quad (30.1)$$

$$\ddot{\Phi}_p(t, \mathbf{q}, \dot{\mathbf{q}}, \ddot{\mathbf{q}}) \equiv \mathbf{J} \left(\sum_{i=1}^{n_{dof}^{(1,5)}} \left(\mathbf{B}_i^{(1,5)} \tilde{\mathbf{r}}_E^{(1,5)} \ddot{\mathbf{q}}_i^{(1,5)} + \sum_{j=1}^{n_{dof}^{(1,5)}} \mathbf{B}_{i,j}^{(1,5)} \tilde{\mathbf{r}}_E^{(1,5)} \dot{\mathbf{q}}_i^{(1,5)} \dot{\mathbf{q}}_j^{(1,5)} \right) + \mathbf{B}^{(P)} \ddot{\hat{\mathbf{r}}}_E^{(P)}(t) \right) = \mathbf{0}. \quad (30.2)$$

From Eqs (30.1) and (30.2) and taking into account (24.1), (24.2), the following components can be identified:

$$\Phi_{p,q} = \begin{bmatrix} \mathbf{J} \mathbf{B}_1^{(1,5)} \tilde{\mathbf{r}}_E^{(1,5)} & \dots & \mathbf{J} \mathbf{B}_i^{(1,5)} \tilde{\mathbf{r}}_E^{(1,5)} & \dots & \mathbf{J} \mathbf{B}_{n_{dof}^{(1,5)}}^{(1,5)} \tilde{\mathbf{r}}_E^{(1,5)} \end{bmatrix}, \quad (31.1)$$

$$\dot{\Phi}_{p,q} \dot{\mathbf{q}} = \sum_{i=1}^{n_{dof}^{(1,5)}} \sum_{j=1}^{n_{dof}^{(1,5)}} \mathbf{B}_{i,j}^{(1,5)} \tilde{\mathbf{r}}_E^{(1,5)} \dot{\mathbf{q}}_i^{(1,5)} \dot{\mathbf{q}}_j^{(1,5)}, \quad (31.2)$$

$$\mathbf{g}_p(t) = \mathbf{J} \mathbf{B}^{(P)} \ddot{\mathbf{r}}_E^{(P)}(t). \quad (31.3)$$

When the model with a fixed base is considered, additional constraint equations need to be defined as

$$\Phi_b(\mathbf{q}^{(b)}) \equiv \mathbf{q}^{(b)} = \mathbf{0}. \quad (32)$$

Finally, equations of motion (26) for the OLKC manipulator model have the form

$$\begin{bmatrix} \mathbf{M}_{i_c} + \mathbf{M}_{d_c} \\ \Phi_{p,q} \\ \Phi_{b,q} \end{bmatrix} \ddot{\mathbf{q}} = \begin{bmatrix} \mathbf{h}_{i_c} + \mathbf{Q}_{i_c} + \Delta \mathbf{Q}_{i_c} + \mathbf{h}_{d_c} + \mathbf{Q}_{d_c} + \Delta \mathbf{Q}_{d_c} \\ -\dot{\Phi}_{p,q} \dot{\mathbf{q}} - \ddot{\mathbf{g}}_p(t) - 2\alpha \dot{\Phi}_p - \beta^2 \Phi_p \\ \mathbf{0} - 2\alpha \dot{\Phi}_b - \beta^2 \Phi_b \end{bmatrix}, \quad (33)$$

where α , β - the Baumgarte method parameters for solution stabilization, $\Phi_{b,q} = [\mathbf{I}_{6 \times 6} \quad \mathbf{0}]$ - the $6 \times n_{dof}^{(1,5)}$ constraint matrix resulting from (30.1), (30.2) $\mathbf{I}_{6 \times 6}$ - identity matrix. In the case when the base support is flexible the constraint (32) and its derivatives are not included into (33).

In Table 1 mass parameters of the manipulator are provided. Friction parameters, stiffness, damping and other parameters are gathered in Table 2.

Table 1 Pseudo-inertia matrices of the manipulator links

Link (α, i)	Pseudo-inertia matrix $\mathbf{H}^{(\alpha, i)}$
Base (b)	$\begin{bmatrix} 26.003 \text{ kg m}^2 & 0 & 0 & 39.005 \text{ kg m} \\ 0 & 6.501 \times 10^{-4} \text{ kg m}^2 & 0 & 0 \\ 0 & 0 & 6.501 \text{ kg m}^2 & 0 \\ 39.005 \text{ kg m} & 0 & 0 & 78.010 \text{ kg} \end{bmatrix}$
Link (1,1)	$\begin{bmatrix} 62.085 \text{ kg m}^2 & 0 & 0 & 93.128 \text{ kg m} \\ 0 & 3.371 \text{ kg m}^2 & 0 & 0 \\ 0 & 0 & 3.371 \text{ kg m}^2 & 0 \\ 93.128 \text{ kg m} & 0 & 0 & 186.257 \text{ kg} \end{bmatrix}$
Link (1,2)	$\begin{bmatrix} 0.268 \text{ kg m}^2 & 0 & 0 & 0.804 \text{ kg m} \\ 0 & 4.693 \times 10^{-5} \text{ kg m}^2 & 0 & 0 \\ 0 & 0 & 3.955 \times 10^{-4} \text{ kg m}^2 & 0 \\ 0.804 \text{ kg m} & 0 & 0 & 3.217 \text{ kg} \end{bmatrix}$

Link (1,3)	$\begin{bmatrix} 5.792 \times 10^{-3} \text{ kg m}^2 & 0 & 0 & 0.289 \text{ kg m} \\ 0 & 2.816 \times 10^{-5} \text{ kg m}^2 & 0 & 0 \\ 0 & 0 & 2.373 \times 10^{-4} \text{ kg m}^2 & 0 \\ 0.289 \text{ kg m} & 0 & 0 & 1.931 \text{ kg} \end{bmatrix}$
Link (1,4)	$\begin{bmatrix} 1.464 \text{ kg m}^2 & 0 & 0 & 5.489 \text{ kg m} \\ 0 & 6.862 \times 10^{-2} \text{ kg m}^2 & 0 & 0 \\ 0 & 0 & 6.862 \times 10^{-2} \text{ kg m}^2 & 0 \\ 5.489 \text{ kg m} & 0 & 0 & 27.448 \text{ kg} \end{bmatrix}$
Link (1,5)	$\begin{bmatrix} 3.676 \text{ kg m}^2 & 0 & 0 & 11.028 \text{ kg m} \\ 0 & 3.970 \times 10^{-2} \text{ kg m}^2 & 0 & 0 \\ 0 & 0 & 3.970 \times 10^{-2} \text{ kg m}^2 & 0 \\ 11.028 \text{ kg m} & 0 & 0 & 44.114 \text{ kg} \end{bmatrix}$

Table 2 Parameters assumed for the manipulator and the mechanism models in simulations

Parameter	Symbol	Manipulator	Mechanism
Static friction coefficient	$\mu_s^{(\alpha,i)}$	0.2	
Kinetic friction coefficient	$\mu_d^{(\alpha,i)}$	0.1	
Stiffness coefficient, m^{-1}	$\sigma_0^{(\alpha,i)}$	100	
Damping coefficient, s m^{-1}	$\sigma_1^{(\alpha,i)}$	1	
Viscous friction coefficients, s m^{-1}	$\sigma_2^{(\alpha,i)}$	0	
Stribeck velocity, ms^{-1} or rad s^{-1}	$\dot{q}_{Stribeck}^{(a,i)}$	0.001 for prismatic joints	
		0.0175 for revolute joints	
Young modulus, Nm^{-2}	E	2.1×10^{11}	
Shear modulus, Nm^{-2}	G	0.8×10^{11}	
Number of sdes of the flexible link	$n_{sde_l}^{(\alpha,i)}$	7	
Stiffness matrix of the flexible link's sde, Nm rad^{-1}	$\mathbf{S}_{sde_l}^{(\alpha,i,j)}$	$\text{diag} \begin{pmatrix} 2.098 \times 10^6 \\ 5.333 \times 10^6 \\ 1.741 \times 10^5 \end{pmatrix}$	$\text{diag} \begin{pmatrix} 35126 \\ 91875 \\ 330.75 \end{pmatrix}$

Number of sdes of the base support	$n_e \times n_{sde_s}^{(i)}$	10×10	
Stiffness matrix of the base support's sde, $N\ m^{-1}$	$S_{sde_s}^{(i,j)}$	$\text{diag} \begin{pmatrix} 10^3 \\ 10^4 \\ 10^3 \end{pmatrix}$	$\text{diag} \begin{pmatrix} 10^2 \\ 10^3 \\ 10^2 \end{pmatrix}$

The influence of the base support stiffness, link's flexibility and friction in joints in the programmed motion performed by the manipulator is analyzed next. Also, forces acting in the joints are inspected. To facilitate the description of subsequent simulation results, the following notation shown in Fig. 6 is introduced. The 4th Runge-Kutte scheme with a constant step size equal to 1×10^{-5} s is applied to integrate the dynamic equations of motion. The Baumgarte method parameters for solution stabilization are selected to be $\alpha = 1 \times 10^3$, $\beta = 1 \times 10^2$.

Fig. 6. The scheme of notation applied in simulation results description

Fig. 7 shows the trajectory and time courses of the end-effector. The starting point of the effector motion differs from the assumed point A . The elimination of disturbances resulting from an arbitrary initial configuration of the manipulator end effector is obtain during its transition from the starting point $*$ to B . After this time, the effector moves along the edges of the ABCD rectangle. The effector is stopped at the corner points and then moves to the next point. For all analyzed cases, during motion from the point B to points C, D, A , violation of the program constraints does not exceed the value of 1×10^{-4} m.

Fig. 7. The calculated trajectory of the end-effector and time courses of its coordinates

Time courses of the joint coordinates are presented in Fig. 8. These results show that the most significant influence on the manipulator motion comes from the flexibility of the base support and friction in joints. Comparing the plots for the manipulator with the rigid link (1,2) and ideal joints ($S_r_L_r_J_{nf}$) with those for the flexible link (1,2) and frictionless joints ($S_r_L_{fl}_J_{nf}$), one can conclude that they are close to each other.

Fig. 8. Time courses of the manipulator joint coordinates

Fig. 9 displays plots of the time courses of force magnitudes acting in the joints. It can be seen that while the flexibility of the links does not have major impact on the manipulator motion, it has a significant effect on the forces generated in the joints. Fig. 10 allows us to analyze RMS values of the particular components of the joint forces.

Fig. 9. Time courses of the force magnitudes acting in each joint

Fig. 10. RMS values of the axial and normal forces acting in the joints

Analyzing the RMS values calculated for the axial and normal forces it can be observed that the highest values of the RMS are obtained for the case $S_{r_L_fl_J_nf}$, i.e. the link (1,2) is flexible, the support is rigid and friction in joints is neglected. The RMS values determined for this case are up to 8 times bigger than the others. Thus, it can be concluded that the link flexibility, in this case, does not have major impact on the manipulator motion, but may cause a significant increase of the joint loads.

The presented analysis of the system performance in programmed motion enables detailed verification and motion planning for constrained tasks.

3.2. The RPSUP mechanism model

The RPSUP (R-revolute, P-prismatic, S-spherical, U-universal, P-prismatic joint) mechanism is the second example and it is the CLKC system model (Fig. 11). The goal of the simulation is to track programmed motion by the end effector E, which is a rectangular trajectory located on the plane $\mathbf{y}^{(P)}\mathbf{z}^{(P)}$ that is inclined to $\mathbf{x}^{(0)}\mathbf{z}^{(0)}$ plane at the angle $\alpha = 45^\circ$. This trajectory reflects some work regime on the end effector. The transformation matrix from the frame $\{P\}$ to the global reference frame has the form:

$$\mathbf{B}^{(P)} = \begin{bmatrix} \cos \alpha & -\sin \alpha & 0 & x_{org}^{(P)} \\ \sin \alpha & \cos \alpha & 0 & y_{org}^{(P)} \\ 0 & 0 & 1 & z_{org}^{(P)} \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad (34)$$

where $x_{org}^{(P)}$, $y_{org}^{(P)}$, $z_{org}^{(P)}$ are coordinates of the position vector $\mathbf{r}_{org}^{(P)}$ describing position of the frame $\{P\}$ with respect to the global one. It is assumed that $x_{org}^{(P)} = 0.0384 \text{ m}$, $y_{org}^{(P)} = 0.2335 \text{ m}$, $z_{org}^{(P)} = 0 \text{ m}$.

In order to formulate the dynamic equations of motion of the mechanism using the CoPCoD algorithm, first it is necessary to remove the kinematic loop using the cut-joint technique. It is assumed that the cut-joint is located at the revolute joint R connecting the flexible link (1,3) with the slider (2,1). Local coordinate systems are assigned to the links as shown in Fig. 12. As a result, two systems of the OLKC types are obtained, for which the generalized coordinates vector results as:

$$\mathbf{q} = \left[\mathbf{q}^{(b)T} \underbrace{\psi^{(1,1)} \quad x^{(1,2)} \quad \psi^{(1,3,0)} \quad \theta^{(1,3,0)} \quad \varphi^{(1,3,0)} \quad \mathbf{q}_f^{(1,3)T}}_{\mathbf{q}^{(1)T}} \underbrace{z^{(2,1)} \quad \psi^{(2,2)}}_{\mathbf{q}^{(2)T}} \right]^T, \quad (35)$$

where: $\mathbf{q}^{(c)}|_{c=1,2}$ is a vector of generalized coordinates of c sub-chain, $\mathbf{q}_f^{(1,3)}$ is a vector containing generalized coordinates of the flexible coupler (1,3).

Fig. 11. The RPSUP mechanism with the programmed trajectory

Next, applying the CoPCoD algorithm presented in the paper, the dynamic equations of motion of the mechanism are obtained as:

$$\begin{bmatrix} \mathbf{M}_{i_c} + \mathbf{M}_{d_c} \\ \Phi_{p,q} \\ \Phi_{k,q} \\ \Phi_{b,q} \end{bmatrix} \ddot{\mathbf{q}} = \begin{bmatrix} \mathbf{h}_{i_c} + \mathbf{Q}_{i_c} + \Delta \mathbf{Q}_{i_c} + \mathbf{h}_{d_c} + \mathbf{Q}_{d_c} + \Delta \mathbf{Q}_{d_c} \\ -\dot{\Phi}_{p,q} \dot{\mathbf{q}} - \ddot{\mathbf{g}}_p(t) - 2\alpha \dot{\Phi}_p - \beta^2 \Phi_p \\ -\dot{\Phi}_{k,q} \dot{\mathbf{q}} - 2\alpha \dot{\Phi}_k - \beta^2 \Phi_k \\ \mathbf{0} - 2\alpha \dot{\Phi}_b - \beta^2 \Phi_b \end{bmatrix}, \quad (36)$$

Fig. 12. Local frames assigned to the mechanism links

Equations (36) are additionally supplemented by closing constraint equations formulated for the revolute cut-joint R. In simulations, as in the case of the manipulator, the influence of the base support flexibility, the coupler flexibility, friction in the joints on the generated programmed motion courses and the forces acting in the joints are analyzed. Mass parameters of the mechanism links applied in simulation tests are presented in Table 3. Other parameters are gathered in Table 2.

Table 3 Pseudo-inertia matrices of the mechanism links

Link (α, i)	Pseudo-inertia matrix $\mathbf{H}^{(\alpha, i)}$
Base (b)	$\begin{bmatrix} 0.812 \text{ kg m}^2 & 0 & 0 & 2.437 \text{ kg m} \\ 0 & 2.031 \times 10^{-5} \text{ kg m}^2 & 0 & 0 \\ 0 & 0 & 0.203 \text{ kg m}^2 & 0 \\ 2.437 \text{ kg m} & 0 & 0 & 9.751 \text{ kg} \end{bmatrix}$
Crank $(1, 1)$	$\begin{bmatrix} 0.401 \text{ kg m}^2 & 0 & 0 & 0.752 \text{ kg m} \\ 0 & 1.392 \times 10^{-4} \text{ kg m}^2 & 0 & 0 \\ 0 & 0 & 1.392 \times 10^{-4} \text{ kg m}^2 & 0 \\ 0.752 \text{ kg m} & 0 & 0 & 1.882 \text{ kg} \end{bmatrix}$

Slider (2,1)	$\begin{bmatrix} 5.201 \times 10^{-4} \text{ kg m}^2 & 0 & 0 & 0 \\ 0 & 1.3002 \times 10^{-6} \text{ kg m}^2 & 0 & 0 \\ 0 & 0 & 1.3002 \times 10^{-6} \text{ kg m}^2 & 0 \\ 0 & 0 & 0 & 0.156 \text{ kg} \end{bmatrix}$
Cylinder (2,2)	$\begin{bmatrix} 4.188 \times 10^{-6} \text{ kg m}^2 & 0 & 0 & 0 \\ 0 & 4.188 \times 10^{-6} \text{ kg m}^2 & 0 & 0 \\ 0 & 0 & 8.935 \times 10^{-5} \text{ kg m}^2 & 0 \\ 0 & 0 & 0 & 0.107 \text{ kg} \end{bmatrix}$
Slider (1,2)	$\begin{bmatrix} 2.206 \times 10^{-2} \text{ kg m}^2 & 0 & 0 & 0.110 \text{ kg m} \\ 0 & 1.838 \times 10^{-5} \text{ kg m}^2 & 0 & 0 \\ 0 & 0 & 1.838 \times 10^{-5} \text{ kg m}^2 & 0 \\ 0.110 \text{ kg m} & 0 & 0 & 0.735 \text{ kg} \end{bmatrix}$
Coupler (1,3)	$\begin{bmatrix} 4.875 \times 10^{-2} \text{ kg m}^2 & 0 & 0 & 0.146 \text{ kg m} \\ 0 & 4.388 \times 10^{-7} \text{ kg m}^2 & 0 & 0 \\ 0 & 0 & 1.219 \times 10^{-4} \text{ kg m}^2 & 0 \\ 0.146 \text{ kg m} & 0 & 0 & 0.585 \text{ kg} \end{bmatrix}$

Runge-Kutta method is used to integrate dynamic equations (36) with a constant step size $h = 1 \times 10^{-5} \text{ s}$. The applied Baumgarte method coefficient values are $\alpha = 500$, $\beta = 50$. Initial conditions for the dynamic analysis are obtained from solving the statics case. Fig. 13 and 14 show the obtained end-effector E trajectory and the corresponding time courses of the end-effector coordinates.

Fig. 13. Motion along the programmed trajectory of the end-effector

Fig. 14. Time courses of the end-effector coordinates in programmed motion

Analyzing the solutions obtained from statics, it can be noticed that the application of the base support flexibility leads to the effector displacement with respect to the point A . Moreover, the coupler flexibility has no major effect on the effector position change at the initial time. Initial arbitrary location of the end effector is eliminated during its transition from the point A to B , and then the mechanism moves according to the programmed constraints. Some distortions of the trajectory for y -component of the end-effector displacement can be observed for the case of $S_r-L_r-J_{nf}$, in which the base support and the coupler are rigid and all joints are treat as ideal. These distortions are even greater at the velocity and acceleration levels (Fig. 15).

Fig. 15. Time courses of the y -components of velocity and acceleration of the end effector

Fig. 16 displays magnitudes of the constraint violations for the kinematic and programmed constraints. The plots are presented using logarithmic scale to obtain better resolution of the results. Significantly large violations of the kinematic constraints can be observed for the case of $S_r_L_r_J_{nf}$ ($\|\tilde{\Phi}_k\| = 4.39 \text{ cm}$ for $t = 2.014 \text{ s}$), despite the use of the Baumgarte stabilization method with different parameter settings to eliminate these errors. However, these violations of the programmed constraints are at the acceptable level. One should also pay attention to the different nature of the time courses of violations of the programmed and kinematic constraints. Violations of the programmed constraints, except in the first case, are linear (in the sense of a logarithmic scale), while violations of the kinematic constraints are strongly non-linear. Fig. 17 shows times courses of joint coordinates for the analyzed cases.

Fig. 16. Constraint violations at the cut-joint R presented in the joint local frame

Fig. 17. Time courses of the displacements of the joint coordinates

Examining the calculated time courses, it can be seen that plots obtained for cases with rigid base support are close to each other. Also, the support flexibility impacts significantly courses of the joint coordinates (cases $S_{fl}-L_r-J_{nf}$, $S_{fl}-L_r-J_f$, $S_{fl}-L_{fl}-J_f$). Friction in joints also has noticeable impact on the overall mechanism motion. The flexibility of the coupler has the least influence on motion of individual links of the system. It is interesting because analyzing deformations of the coupler (Fig. 18) it can be concluded that they are large (for $t = 2$ s deformation of the coupler is around 4 cm for the cases with rigid support). On the other hand, links flexibility has visible impact on forces acting in joints. Fig. 19 shows RMS values calculated for the normal and axial forces acting in each joint of the mechanism.

Fig. 18. Coupler deformations at selected time instants

Fig. 19. RMS values of forces acting in the joints

It can be seen that the highest values of the normal and axial forces are obtained for the case in which the coupler (1,3) is flexible. Links flexibility has significant impact on loading of the mechanism and cannot be neglected in numerical simulations.

4. Conclusions

The general modeling procedure for multibody systems with serial structures of the OLKC and CLKC types subjected to the programmed constraints is presented in the paper. The proposed computational procedure, referred to as CoPCoD, provides an efficient algorithm for dynamics modeling and enables taking into account nonlinearities due to flexibility of the base and links and friction in joints. It is based on the joint coordinates formalism and homogeneous transformation matrices. The CoPCoD produces the constrained dynamics, the so called reference dynamics, which enables detailed analysis of the system performance in the desired programmed motion. The underlying dynamics is derived using the GPME algorithm. In simulation tests presented in the paper, for both OLKC and CLKC system models, the influence of the model structure and its properties on the programmed motion were studied. Numerical experiments show that the CoPCoD allows obtaining solutions of the reference dynamics effectively. The computer generation of the model dynamics with the CoPCoD is suitable for

various system structures. The presented results indicate how the base flexibility, friction in joints, adoption of rigid or flexible link models impacts motion of the system model. For example, link flexibility can affect motion of the mechanism if the deformation of the link is very large. In our example, the maximum deformations of flexible links are about 10% of the length of links, and their influence on motion of the analyzed systems are negligible. However, the link flexibility has a significant effect on the forces acting in the joints, and cannot be ignored in numerical simulations, even when deformations are small. The future works planned to CoPCoD upgrades to make it more powerful, will be focused on looking for reference dynamics for systems for which the programmed constraints are met with some tolerance, and additional constraints resulting from e.g. design requirements needs to be satisfied.

Conflict of interest: The authors declare that they have no conflict of interest.

Funding: Open access funding provided by Warsaw University of Technology and University of Bielsko-Biala

Data availability: The C++ codes and the datasets generated during the current studies are available from the corresponding author on reasonable request.

References

1. Schiehlen W (1997) Multibody system dynamics: roots and perspectives, *Multibody Syst Dyn* 1(2): 149-188, doi:10.1023/A:1009745432698.
2. Shabana A (1997) Flexible multibody dynamics: review of past and recent developments, *Multibody Syst Dyn* 1(2): 189-222, doi:10.1023/A:1009773505418.
3. De Luca A, Siciliano B (1991) Closed-form dynamic model of planar multilink lightweight robots, *IEEE Trans Syst Man Cybern.* 21(4): 826-839, doi:10.1109/21.108300.
4. Mohan A, Saha S (2009) A recursive, numerically stable, and efficient simulation algorithm for serial robots with flexible links, *Multibody Syst Dyn.* 21(1): 1-35, doi:10.1007/s11044-008-9122-6.
5. Krauss RW, Book WJ (2010) Transfer matrix modeling of systems with non-collocated feedback, *J Dyn Syst Meas Control* 132(6), 061301, doi:10.1115/1.4002476.
6. Theodore RJ, Ghosal A (1995) Comparison of the assumed modes and finite element models for flexible multilink manipulators, *Int J Robot Res* 14(2): 91-111.
7. Chebbi J, Dubanchet V, Perez Gonzalez JA, Alazard D (2016) Linear dynamics of flexible multibody systems: a system-based approach, *Multibody Syst Dyn.*, Springer Verlag, 10.1007/s11044-016-9559-y, hal-01405184.
8. Leckie F, Pestel E (2016) Transfer-matrix fundamentals, *Int J Mech Sci* 2: 137-167.
9. Li H, Zhang X (2016) A Method for Modeling Flexible Manipulators: Transfer Matrix Method with Finite Segments, *International Journal of Computer and Information Engineering* 10(6): 1086-1093.
10. Rui X, Wang G, Lu Y, Yun L (2008) Transfer matrix method for linear multibody system, *Multibody Syst Dyn* 19(3): 179-207, doi:10.1007/s11044-007-9092-0.
11. Kitis L, Lindenberg R (1990) Natural frequencies and mode shapes of flexible mechanisms by a transfer matrix method, *Finite Elem Anal Des* 6(4): 267-285, doi:10.1016/0168-874X(90)90020-F.

12. Jarzębowska E, Augustynek K, Urbaś A (2017) Computational reference dynamical model of a multibody system with first order constraints. Proc. ASME 2017 Int Design Eng Techn Conf and Comp Inform Eng Conf, IDETC/CIE 2017, August 6-9, Cleveland, Ohio, USA.
13. Hairer E., Nørsett SP, Wanner G (1993) Solving Ordinary Differential Equations II: Stiff and Differential-Algebraic problems, Springer, Berlin.
14. Ascher UM, Petzold LR (1993) Stability of Computational Methods for Constrained Dynamics Systems, SIAM J Sci Comput 14(1): 95-120.
15. Schiehlen W (1991) Computational Aspects in Multibody System Dynamics, Computer Methods in Applied Mechanics and Engineering 90: 569-582.
16. Schiehlen W, Guse N, Seifried R (2006) Multibody Dynamics in Computational Mechanics and Engineering Applications, Comput Methods Appl Mech Eng 195:5509-5522.
17. Marques F, Souto AP, Flores P (2017) On the Constraints Violation in Forward Dynamics of Multibody Systems, Multibody Syst Dyn 39: 385-419, doi: 10.1007/s11044-016-9530-y.
18. Aghili F (2008) Dynamics and control of constrained mechanical systems in terms of reduced quasi-velocities, Proc. 2008 IEEE Int. Conf. Robotics and Automation, Pasadena, CA, 1225-1230.
19. Xiaoping Y., Nilanjan S (1998) Unified formulation of robotic systems with holonomic and nonholonomic constraints. IEEE Trans. Robot. Automat., 14(4): 640 - 650.
20. Udwadia F., Kalaba R., Hee-Chang E (1997) Equations of motion for constrained mechanical systems and extended D'Alembert's principle, Quarterly of Applied Mathematics LV (2): 321-331.
21. Udwadia F., Wanihanon T (2013) On general nonlinear constrained mechanical systems, Numerical Algebra, Control and Optimization, 3(3): 425-443, doi: 10.3934/naco.2013.3.425.
22. Udwadia F., Mylapilli H (2015) Constrained motion of mechanical systems and Tracking control of nonlinear systems: connections and closed-form results, Nonlinear Dynamics and Systems Theory 15(1): 73-89.
23. Liu X., Zhen S., Huang K., Zhao H., Chen Y-H., Shao K (2015) A systematic approach for designing analytical dynamics and servo control of constrained mechanical systems, IEEE/CAA Journal of Automatica Sinica, 2 (4): 382-384.
24. Lot R, Dalio M (2004) Symbolic Approach for Automatic Generation of the Equations of Motion of Multibody Systems, Multibody Syst Dyn 12: 147-172.
25. Koutsovasilis P., Beitelschmidt M (2007) Simulation of constrained mechanical systems, proceedings in Applied Mathematics and Mechanics, doi: 10.1002/pamm.200700900.
26. Schutte A, Udwadia F (2011) New Approach to the Modeling of Complex Multibody Dynamical Systems, J Appl Mech 78(2), doi:10.1115/1.4002329.
27. Balseiro P., Marrero J.C., De Diego M.D., Padron E (2010), A unified framework for mechanics – Hamilton-Jacobi equation and applications, London Math. Society, *Nonlinearity*, 23, 1887-1918.
28. Sandino L.A., Bejar M., Ollero A (2013), A Survey on methods for elaborated modeling of the mechanics of a small-size helicopter; analysis and comparison, *J. Intell. Robot. Syst.*, 72, 219-238.
29. Sira-Ramirez H., Agrawal S (2004) Differentially Flat Systems, CRC Press.
30. Yuh J (2000) Design and control of autonomous underwater robots: A survey, *Autonomous Robots*, 8: 7-24.
31. Yu Cheng, Dong Ye, Zhaowei Sun (2018) Attitude-constrained reorientation control for spacecraft based on extended state observer, *Advances in Mechanical Engineering*, Vol. 10(8) 1-12, doi: 10.1177/1687814018794525

32. Ellery, A. (2004) An engineering approach to the dynamic control of space robotic on-orbit servicers. *Proc Inst Mech Eng, Part G: J Aerospace Eng.*, Vol. 218 79-98.
33. Jarzębowska E and Pietrak K (2014) Constrained mechanical systems modeling and control: A free-floating space manipulator case as a multi-constrained system. *Robotics and Autonomous Systems* 62: 1353-1360.
34. Koh K.C., H. S. Cho (1999) A smooth path tracking algorithm for wheeled mobile robotics with dynamic constraints. *Journal Intell.Robot. System.*, 24: 367-385.
35. Oriolo G., Nakamura Y (1991) Control of mechanical systems with second-order nonholonomic constraints: underactuated manipulators. *Proc. of 30th Conf. Decision Contr.*, 2398-2403.
36. Scheuer A., Laugier Ch (1998) Planning sub-optimal and continuous-curvature paths for car-like robots. *Proc. 1998 IEEE/RSJ Intl. Conf. Intell. Robot. Syst.*: 25-31.
37. Shekl A.M., Lumelsky V.L (1998) Motion planning for nonholonomic robots in a limited workspace. *Proc. 1998 IEEE/RSJ Intl. Conf. Intell. Robot. Syst.*: 1473 – 1478.
38. Singh S.K., Leu M.C (1991) Manipulator Motion Planning in the Presence of Obstacles and Dynamic Constraints. *Int. Journal of Robotics Research*, 10(2): 171-187.
39. Jarzębowska E, Augustynek K, Urbaś A (2018) Development of a computational based reference dynamics model of a flexible link manipulator, *Springer Proc. Mathematics & Statistics series, Dynamical systems in Applications* 249: 169-180.
40. Jarzębowska E, Augustynek K, Urbaś A (2018) Automated generation of reference dynamical models for constrained robotic systems in the presence of friction and damping effects, *Concurrency and Computation: Practice and Experience*, doi:10.1002/cpe.4452.
41. Jarzębowska E (2005) Dynamics Modeling of Nonholonomic Mechanical Systems: Theory and Applications, *Nonlinear Analysis* 63(5-7): 185-197.
42. Jarzębowska E (2008) Quasi-Coordinates based Dynamics Modeling and Control Design for Nonholonomic Systems, *Nonlinear Analysis* 16(16): 1741-1754.
43. Jarzębowska E.(2016a) An underwater inertia-based propelled vehicle dynamics and performance, *Proc. XXIV International Conference of Theoretical and Applied Mechanics*, 21-26 August 2016, Montreal, Canada.
44. Jarzębowska E., (2016b) Tracking control design for variable mass and configuration robotic systems, *Proc. ASME 2016 Dynamic Systems and Control Conference*, ASME Digital Collection.
45. Banaszuk A., Mehta P.G., Hagen G (2006) The role of control in design: from fixing problems to the design of dynamics, *International IFAC Symposium on Advanced Control of Chemical Processes*, Gramado, Brazil – April 2-5: 913-928.
46. Wittbrodt E, Szczotka M, Maczyński A, Wojciech S (2013) Rigid Finite Element Method in Analysis of Dynamics of Offshore Structures, *Ocean Engineering & Oceanography*, Springer, Berlin-Heidelberg.
47. Augustynek K, Urbaś A (2017) Two Approaches of the Rigid Finite Element Method to Modelling the Flexibility of Spatial Linkage Links, *Proc. of the ECCOMAS Thematic Conference on Multibody Dynamics*, June 19-22, Prague, Czech Republic
48. Canudas de Wit C, Ollson H, Åström KJ, Lischinsky P (1995) A new model for control of systems with friction, *IEEE Transactions on Automatic Control* 40(3): 419-425.
49. Craig J.J (1989) Introduction to robotics. Mechanics and control, Addison-Wesley Publishing Company, Inc.